Migración: Copiar archivos directamente de un hosting a otro

Luciano Martínez

Uno de los dolores de cabeza de muchos de nosotros al migrar sitios de un hosting a otro surge cuando hay que mover los archivos. Hablando de hosting compartido, el proceso normal se resume en comprimir los archivos, descargarlos y subirlos al nuevo hosting. Historia muy bonita hasta que te encuentras con un sitio de más de 2gb y tu conexión de banda ancha funciona peor que un dialUp de 56kbps. Unas cuantas líneas de código pueden ser la solución.

Una de los problemas de usar el siguiente código, puede ser que tu hosting tenga alguna restricción, y tengas que elegir el método clásico.

Pero vamos al grano. Para este ejemplo, nombraremos como ORIGEN al servidor desde el que deseamos copiar los archivos y DESTINO a que vamos a copiar la información.

¿Qué necesitamos?

- Los datos de acceso FTP de ambos servidores
- El archivo o conjunto de archivos ORIGEN en un solo archivo comprimido que puede ser ZIP o JPA.
- Si se trata de un respaldo del sitio ORIGEN que creaste con Akeeba Backup, es mejor que muevas el archivo a una carpeta diferente a aquella que por defecto se usa para guardar los respaldos.

Procedimiento.

Existen dos maneras de hacerlo. La primera consiste en copiar el archivo en un solo evento, es decir que solo se realizará una solicitud al servidor en la que se copiará todo el archivo. El problema con este método es que tu servidor DESTINO puede tener restricciones (como el tiempo de ejecución) que evite copiar el archivo completo o

en buen estado sobre todo si se trata de un archivo muy grande.

La segunda forma, que es la que tratamos en este artículo, realiza la copia de forma asíncrona, es decir que el archivo se copia mientras se pueden continuar realizando otras tareas. Así que una vez que ejecutes el código que explico a continuación, solo necesitas un poco de paciencia, mucha menos de la que necesitarías si primero bajas el archivo a tu computadora y después lo subes al nuevo hosting (a menos que tengas una súper conexión de internet, que al menos por las regiones mexicanas, no es muy frecuente). Procedamos:

PASO 1

Ingresa por FTP a tu hosting DESTINO. Puedes hacerlo, por ejemplo, con *FileZilla* o el mismo administrador de archivos que debe tener el panel de control de tu hosting.

PASO 2

Crea en tu hosting DESTINO una carpeta llamada 'descarga' y dale permisos 777. (NOTA ACLARATORIA)

PASO 3

En la carpeta 'descarga' que creaste en el paso anterior, inserta un nuevo archivo llamado 'download.php' e inserta el siguiente código:

Primero creamos un archivo para escribir el DESTINO

```
$file = 'respaldoxxxxxx.jpa'; //o zip o lo que
sea.
```

Y lo ponemos listo para escribir. *fopen()* creara el archivo con el nombre ($file) y lo dejará listo para su escritura gracias al parámetro 'w'.

```
$fp = fopen($file,
'w');
```

Ahora vamos a realizar una conexion FTP a nuestro ORIGEN. Primero iniciamos dicha conexión con la función *ftp_connect()* a la cuál le indicaremos la URL o IP del host. Después utilizamos ftp_login() para acceder al servidor indicando 3 argumentos: la variable con la conexión, el nombre de usuario y la contraseña.

```
$conn_id =
ftp_connect('ftp.tusitio.com');
```

```
$login_result = ftp_login($conn_id, 'usuario@ misitio.com',
'EstaEsMiContraseña');
```

Ahora iniciamos la descarga.

La función *ftp_get()* se utiliza para obtener un archivo desde un servidor FTP. El problema, como indiqué anteriormente, es que esta función está limitada al tiempo de ejecución de tu hosting, así como al tamaño del

archivo.

Par los fines que nos convienen aquí, utilizaremos la función *ftp_nb_fget()* que hace lo mismo, con la diferencia que el archivo lo obtiene de forma asíncrona así que tu programa puede desempeñar otras operaciones mientras el archivo se está descargando. Esta función devuelve uno de tres valores:

FTP_FAILED .- Cuando hay una falla de conexión, un problema de escritura en el DESTINO, etc.

FTP_MOREDATA .- Significa que la función está solicitando más datos. En seguida veremos como proporcionárselos.

FTP_FINISHED .- Cuando se termina la transferencia.

Los argumentos necesarios para esta función son: el id de la conexion, el nombre del archivo DESTINO la ruta completa del archivo ORIGEN y la constante FTP_BINARY.

```
$ret = ftp_nb_fget($conn_id, $fp, 'public_html/carpeta/respaldoxxxxxx.jpa',
FTP_BINARY);
```

Mientras el resultado que arroje **ftp_nb_fget()** sea **FTP_MOREDATA**, debemos continuar con la descarga con **ftp_nb_continue()**:

```
                                // Continuar la descarga.
while ($ret == FTP_MOREDATA)    $ret =
{                               ftp_nb_continue($conn_id);        }
```

En caso de que se presente algún error:

```
if ($ret == FTP_FAILED) {
echo "Hubo un error al descargar el
archivo...";
exit(1);
}
```

Por último cerramos la conexión FTP.

```
fclose($fp);
?>
```

Con esto terminamos el código, solo resta ejecutarlo: http://hostingdestino.com/descarga/download.php . Son muy pocas líneas como puedes ver.

Notarás que, a menos de que se trate de un error, el código no imprime nada en pantalla así que no te extrañe que después de un momento tu explorador te envíe un error por superar el tiempo de espera. No te preocupes, puedes ir verificando el tamaño del archivo DESTINO y verás éste se sigue descargando.

Alternativamente puedes utilizar, por ejemplo, la función flush() para ir mostrando alguna barra de estado o un mensaje en cada iteración del 'while' o simplemente sentarte a esperar.

Algunos servicios de hosting, por lo general los que son privados (VPS), te ofrecerán alguna solución alternativa

para trasladar tus archivos directamente. Sin embargo, vale la pena probar este método ya que puedes utilizarlo, por ejemplo, para crear una capa FTP en tus desarrollos y para comprender mejor como funciona dicha capa en algunos CMS com Joomla!

Debes tomar en cuenta que es importante tener la información en un lugar seguro, así que en tus próximas vacaciones puedes dejar tu computadora descargando el respaldo en lo que vuelves a casa. Por lo pronto ya te ahorraste timpo y pudiste terminar de migrar la info a tu nuevo hosting.

Si deseas agregar algún comentario, sugerencias u observaciones sobre este código, por favor no dudes en hacerlo. Siempre es muy agradable aprender y es mejor hacerlo todos juntos. Hasta la próxima.

NOTA ACLARATORIA (4 DE NOVIEMBRE 2013)

Si bien, este artículo no se centra en la seguridad, es importante aclarar que el hecho de crear la carpeta DESTINO con permisos 777 debe tomarse como **una sugerencia** que se hace con el objetívo único de que al trasladar el archivo a dicha carpeta se reduzcan las posibilidades de errores de escritura. Por ningún motivo se deben dejar carpetas y/o archivos con permisos 777 que ponen en riesgo total la seguridad de nuestro sitio y todos sus archivos.

Una vez que termines de trasladar tus archivos a la carpeta DESTINO, es importante colocar los permisos nuevamente en 755 para carpetas y 644 para archivos o los permisos que mejor te convengan. Para mayor información sobre la seguridad y permisos dentro de tu instalación de Joomla! y en general para las carpetas y archivos en tu cuenta de hosting, puedes consultar la documentación oficial de Joomla! sobre este asunto en el link: http://docs.joomla.org/Verifying_permissions.